

Programming Strategy to Reduce Energy Consumption of Smartphone Sensors

L. S. Oliveira and F. L. de Mello

Abstract— This paper presents a strategy to reduce the energy demand of network applications running on a smartphone, by specifying regular intervals for the recurrent services required by these applications. Although there are several initiatives for reducing energy consumption on smartphones, there is a bad practice of software developers for mobiles in which the energy consumption of sensors is not considered as a requirement. Therefore, this work proposes a software pattern for developing mobile applications that allows managing sensors' energy consumption. It contributes for the improvement of energy efficiency. In order to test the efficiency of the proposed solution, two tests were conducted: (1) normal consumption, the application was executed without implementing any method to reduce its energy consumption, and (2) low consumption, where the application was executed by applying the schemas used to reduced the demand for energy on the GPS and Wifi sensors. The energy consumption associated with such strategy demanded less than 20% of the energy required by ordinary mobile applications.

Index Terms—energy consumption, mobile devices, Wifi, GPS.

I. INTRODUÇÃO

O tema deste trabalho é maximização do tempo útil da bateria de *smartphones* empregando uma abordagem de programação orientada à redução do consumo de energia. Deste modo, o problema a ser resolvido é a redução da demanda energética de aplicativos de dispositivos móveis baseados em geolocalização através de uma programação melhor elaborada da ativação dos sensores de GPS (*Global Positioning System*) e Wifi (tecnologia IEEE 802.11).

Smartphones têm se tornado dispositivos indispensáveis no cotidiano das pessoas devido à sua natureza multitarefa. Graças ao seu poder de processamento, é muito comum que tais aparelhos sejam utilizados para atividades diversas, tais como: acessar clientes de e-mail, obter dados geográficos, enviar mensagens instantâneas, reproduzir vídeos, executar jogos, dentre outros. Contudo, prolongar a duração do tempo de vida da bateria enquanto mantém o aparelho conectado às diversas funcionalidades de rede é um grande desafio [1,2].

A questão sobre um gerenciamento eficaz dos recursos dos *smartphones* se torna mais evidente ao se considerar uma das principais restrições desses dispositivos: o tamanho.

Com o objetivo de oferecer maior mobilidade e conveniência às pessoas, os *smartphones* precisam possuir um tamanho compacto. Essa restrição traz complicações relacionadas às dimensões de seus componentes, principalmente das baterias, responsáveis por um recurso vital para o funcionamento do aparelho. A capacidade em Ampère hora (Ah) total dessas baterias acaba sendo reduzida de modo a se adequar a menores dimensões físicas, o que acaba influenciando diretamente o tempo de disponibilidade do aparelho. Essa limitação torna-se mais significativa durante o uso recorrente dos sensores do dispositivo. É comum que aparelhos que realizam uso contínuo dos sensores de GPS ou de Wifi possuam a carga total de suas baterias restritas a poucas horas de duração.

Portanto, torna-se necessária a adoção de mecanismos que visem fazer uso consciente dos sensores de Wifi e GPS de modo a reduzir a sua demanda por energia, permitindo maior prolongamento do tempo de vida da bateria. Desse modo, este trabalho propõem o uso de duas estratégias, implementadas diretamente no código-fonte das aplicações *mobile*, que permitirão um eficaz gerenciamento energético desses sensores. A primeira estratégia define os valores mais apropriados para uso do sensor de GPS considerando a taxa de atualização em função do tempo. A segunda estratégia envolve o desenvolvimento de um algoritmo capaz de atrasar a execução de serviços de rede, com o objetivo de reduzir a taxa de transição do sensor entre os estados de *idle* e em uso, e vice-versa.

Oliveira e Oliveira [17] apresentaram um estudo preliminar e prospectivo sobre hipótese de redução no consumo de energia em dispositivos móveis a partir de uma gerência dos sensores Wifi e GPS. Na ocasião, foi realizado um experimento controlando o acionamento de GPS e Wifi, ainda como uma prova de conceito e sem engenharia de software. Os valores das variáveis relacionadas com a configuração de tais sensores foram definidos empiricamente, e uma avaliação pontual foi feita a partir de um estudo de caso. O presente artigo busca formalizar melhor esta iniciativa, fazendo uma abordagem mais rigorosa e aprofundada. Desta forma, é feita uma engenharia com vistas a produzir um padrão de software orientador de implementações desta natureza, são realizadas definições de estratégias de consumo de energia e regras para enquadramento nestas estratégias, além de um experimento estatístico a fim de produzir valores otimizados para as variáveis de configuração dos sensores.

Assim, esse trabalho é estruturado da seguinte forma: na segunda seção apresenta alguns estudos que influenciaram a construção da estratégia descrita neste trabalho; na terceira seção são detalhados os esquemas de programação que compõe esta estratégia de redução de consumo de energia da

Submitted on July, 4, 2018.

Leonardo da Silva Oliveira is bachelor in Computer Science at Veiga de Almeida University and consultant on software development, dev.loliveira@gmail.com

Flávio Luis de Mello (D.Sc.) is associate professor at the Electronics and Computer Department from Polytechnic School at Federal University of Rio de Janeiro, fmello@poli.ufrj.br

bateria dos dispositivos móveis; a quarta seção é voltada para testes experimentais da estratégia ora proposta. Por fim, na última seção, são realizadas conclusões sobre a eficácia da referida estratégia são apontadas algumas sugestões de trabalhos futuros para avançar sobre o tema.

II. TRABALHOS RELACIONADOS

Tarkoma et al. [1] mencionam que alcançar um prolongamento de duração da carga da bateria enquanto o aparelho faz uso de diversas funcionalidades de rede é um grande desafio, o que aponta o dispositivo Wifi como um consumidor significativo de energia. Carrol e Heiser [2] realizaram uma pesquisa capaz de mensurar o consumo de energia dos seguintes componentes do aparelho: a CPU (*Central Processing Unit*), o LCD (*Liquid-Crystal Display*), a RAM (*Random Access Memory*), o GPS, o Wifi, o GSM (*Global System for Mobile communications*), o Bluetooth e dos cartões SD (*Secure Digital*) e flash. Os resultados de sua pesquisa apontaram que os componentes CPU, LCD e os sensores responsáveis pela realização de stream de dados (Wifi, GSM) estão entre os componentes de maior consumo energético no aparelho. Estes dados se tornam ainda mais preocupantes para o contexto de aplicativos baseados em geolocalização contínua, tratados pela presente pesquisa, uma vez que necessitam do uso recorrente dos sensores de Wifi e GPS para obter uma localização precisa do aparelho. Schneider et al. [14] apresentam uma revisão de literatura dedicada especificamente à avaliação e orientação sobre os componentes das baterias de dispositivos móveis e boas formas de uso.

Uma proposta similar, apresentada por Naik e Chavan [7] em 2015, elaborou uma relação entre os componentes dos *smartphones* com seus respectivos consumos de energia em diferentes estados de uso. O componente de Wifi, por exemplo, foi avaliado em três estados de operação distintos: conectado, em *idle* e baixando dados a uma taxa de 4.5Mbps. Os autores notaram que os componentes relacionados à transmissão de dados pela rede (GPS, Wifi), junto com os picos de uso na CPU, são os principais responsáveis pelo consumo de energia nos aparelhos. Este levantamento reforça a proposta de estratégia apresentada no presente artigo de que os componentes Wifi e GPS devem ser controlados para consumir energia de forma menos intensa.

Uma abordagem centrada no Sistema Operacional (SO), apresentada por Roy et al. [3], detalhou a maneira o funcionamento interno do Cinder, um SO projetado para ser executado em dispositivos com limitações de recursos. Sua contribuição principal está na definição de valores limitrofes para disparo dos mecanismos de gerenciamento de energia. Já Li e Halfond [4], apontam um guia rudimentar e pouco quantitativo para auxiliar a comunidade de desenvolvedores a aprimorar a eficiência energética de seus aplicativos, através da adoção de técnicas simples durante o processo de desenvolvimento. Neste sentido, fica clara a necessidade de realizar experimentos para definir valores determinísticos para os mecanismos de gerenciamento de energia.

Outra abordagem, que conseguiu alcançar resultados positivos no tocante à redução do consumo de energia por parte do Wifi, é descrita por Nedeveschi et al. [5]. Sugere-se a operação do sensor utilizando duas abordagens distintas: (1) desligando o sensor de rede quando o mesmo estiver no estado *idle* e (2) reduzindo a frequência de uso do sensor enquanto o mesmo estiver processando os pacotes de rede. Os autores concluíram que foi possível implementar um modelo eficiente para a redução do consumo de energia a uma baixa redução de desempenho na rede. Vicente *et al.* [13] são mais específicos, discutindo inclusive o que seriam boas configurações do dispositivo móvel para uma melhor redução no consumo de energia.

Podlesny [9] elaborou um modelo para aprimoramento do uso de serviços sensíveis ao atraso de rede através da adoção de duas estratégias: (1) a adoção de um protocolo para controle de congestionamento no tráfego da rede e (2) o uso de estratégias de gerenciamento de *buffer* e algoritmos de agendamento, implementados diretamente em roteadores. O trabalho interessante porque induz o uso das filas virtuais utilizadas em uma das estratégias ser apresentada na próxima seção.

Um modelo de gerenciamento de energia adaptativo para o SO Android foi proposto por Couto et al. [10]. O objetivo de sua proposta era elaborar um modelo capaz de realizar um monitoramento das aplicações e a associação entre o consumo de energia por parte das mesmas com seus respectivos segmentos de código. Em parte, este trabalho contribuiu para o planejamento dos experimentos da seção IV do presente artigo.

Xia et al. [6] propuseram um modelo de GPS assistido por uma nuvem com o objetivo de auxiliar o sensor de Wifi a localizar, com maior eficiência, pontos de acesso que estiverem ao alcance do aparelho. Essa ideia remete para a possibilidade de se utilizar dados com qualidades distintas, o que no caso deste trabalho envolve o emprego do GPS assistido para posicionamento.

Ellison et al. [7] apontam uma tendência dos usuários em utilizar *smartphones* como um substituto para aparelhos de GPS dedicados durante viagens. Com base nessa premissa os autores desenvolveram um aplicativo implementando um balanceamento entre prolongamento do tempo de vida da bateria e a qualidade dos dados obtidos pelo aplicativo através do sensor de GPS em conjunto com outros sensores. Na perspectiva do aplicativo, os autores concluíram que uma sinergia entre os sensores disponíveis no aparelho com o GPS contribuiu para uma aquisição de dados de qualidade equivalente e, até mesmo superior, para viagens realizadas em regiões urbanas.

Santana *et al.* [15] também se preocupam com a questão do consumo de energia pelos dispositivos móveis. Ao invés de promoverem um estudo centrado nas aplicações executadas nestes dispositivos, bem como os componentes de hardware utilizados pelos mesmos, eles sugerem transferir esse consumo de energia para servidores na nuvem. Sob esta ótica, os autores apresentam um estudo que avalia o desempenho das aplicações *mobile* que utilizam recursos desta nuvem. Já Diaz

et al. [16] discutem sobre a economia de energia oriunda do mecanismo de acesso ao meio rádio chamada *Long Term Evolution (LTE)*.

III. ESTRATÉGIA PARA O DESENVOLVIMENTO DE SOFTWARE COM CÓDIGO ORIENTADO À REDUÇÃO DO CONSUMO DE BATERIA

NESTA seção, são apresentadas as estratégias para gerenciamento energético dos sensores de Wifi e GPS, implementadas no código-fonte da aplicação. A estratégia relacionada com o Wifi buscou construir uma fila virtual de serviços de rede para restringir a ativação do sensor a intervalos regulares. Já na estratégia de controle do GPS, foram utilizados os mecanismos disponibilizados pelo *Software Development Kit (SDK)* oficial do Android para definir a frequência com que o GPS deveria ser acionado. Para tal foram especificadas duas variáveis representando os filtros de tempo e distância. Um dos objetivos da elaboração destas duas estratégias é criar uma estratégia estrutural e comportamental de desenvolvimento de software, que possa ser seguida por desenvolvedores de aplicações para auxiliá-los a solucionar os problemas recorrentes de gerenciamento eficaz de energia de seus aplicativos.

Os princípios da programação orientada a objetos foram aproveitados durante a implementação das estratégias para redução do consumo de energia pelos sensores de Wifi e GPS. Sob esta ótica, foi construída uma classe base para cada um dos sensores com o objetivo de viabilizar a herança da solução para um eficaz gerenciamento dos mesmos.

No que diz respeito ao gerenciamento do sensor de Wifi foi construída a classe abstrata *NetworkScheduler*, que atua como um agente centralizador para a execução dos serviços de rede na aplicação. Essa classe implementa a interface *Runnable*, provida nativamente pelo Java, e tem por objetivo viabilizar o uso de *threads*. Uma vez que os serviços não serão executados de imediato, há a necessidade de se construir uma fila virtual para armazenar a referência aos mesmos. Essa fila foi construída indiretamente, utilizando o gerenciamento automático de *threads* do SO Android. No algoritmo, a fila de serviços de rede foi construída empregando os componentes *System* e *Thread*. A classe *System* é utilizada para auxiliar o cálculo do tempo e garantir que cada serviço de rede seja executado paralelamente aos demais serviços na fila. Por fim a classe *Thread* foi utilizada para garantir a execução paralela dos serviços, de modo que cada serviço seja executado em sua própria *Thread*. A Fig.1 apresenta o diagrama de componentes relacionado ao algoritmo do agendamento de serviços de rede.

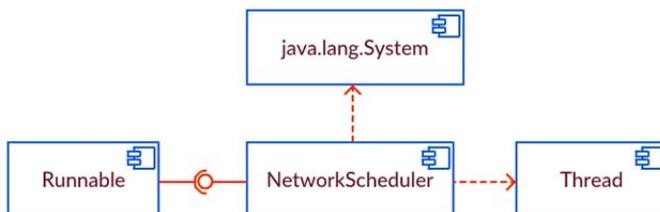


Fig. 1. Diagrama de componentes para *NetworkScheduler*.

O algoritmo de escalonamento implementado na classe *NetworkScheduler* considera o modo de operação de consumo

da aplicação para determinar a frequência com que os serviços de suas subclasses serão executados. Cada serviço adicionado à fila é agendado para ser executado paralelamente aos demais, considerando o valor do tempo de atraso estipulado pela classe. A Fig.2 ilustra a correspondente classe listando os principais atributos e métodos implementados nesse algoritmo.

NetworkScheduler	
NORMAL_CONSUMPTION:	int
LOW_CONSUMPTION:	int
DELAY:	int
NEXT_CALL:	int
scheduleNextCall(task):	void
stopLowerConsumption():	void
switchToLowerConsumption():	void

Fig. 2. Classe *NetworkScheduler*.

Todos os serviços de rede que herdarem de *NetworkScheduler* precisam implementar o algoritmo de suas respectivas responsabilidades no método *run*, que será chamado pela superclasse quando chegar o momento de executar a fila de serviços. Além disso, o tempo máximo de espera para iniciar a execução dos serviços é armazenado por uma variável. Desta forma, essa abordagem adota a estratégia do atraso de rede para reduzir a frequência com que o sensor de Wifi transita entre os estados de *idle* e em uso durante a execução dos serviços. O pseudocódigo para escalonar a tarefa é apresentado a seguir:

```

scheduleNextCall( task ):
  t ← System.currentTimeMillis()

  se t ≤ NEXT_CALL
    scheduleAt ← NEXT_CALL - t
  senão
    scheduleAt ← 0
  NEXT_CALL ← System.currentTimeMillis() + DELAY

  thr ← novo Thread()
  thr.sleep(scheduleAt)
  thr.run( task )
  
```

A segunda estratégia visa gerenciar o uso do sensor de GPS através da especificação dos valores de tempo e distância durante sua inicialização. Compete à classe abstrata *GPSManager* retratar o algoritmo dessa abordagem. A Fig. 3 apresenta um diagrama de componentes para melhor visualização da solução. Nela é possível notar que *GPSManager* centraliza a interação entre os demais componentes envolvidos.

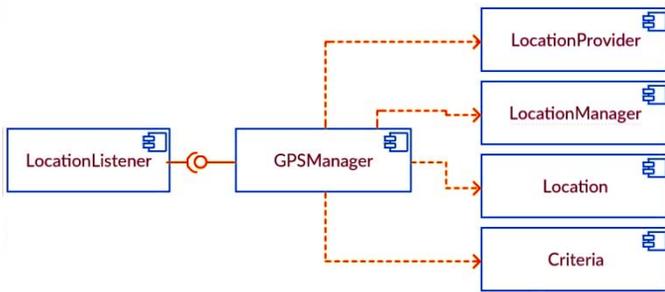


Fig. 3. Diagrama Diagrama de componentes para *GPSManager*.

A interface *LocationListener* é utilizada para garantir a implementação dos métodos pertinentes aos eventos relacionados ao GPS. *LocationManager* é a classe responsável pela ativação do sensor, através do método *requestLocationUpdates*. Por fim, as classes *LocationProvider* e *Criteria* são utilizadas para definir os filtros para seleção do provedor de serviços mais adequado, dentre as seguintes opções: gps, rede e passivo. O provedor “gps” está associado ao dispositivo GPS do equipamento e portanto oferece maior precisão quanto à localização calculada, porém ao custo de maior consumo de energia. O provedor de “rede” refere-se à técnica de GPS assistido, logo, utiliza as torres de telefonia e os pontos de acesso ao alcance do aparelho para inferir sua localização, possuindo precisão e consumo de energia inferiores aos do provedor de “gps”. Por fim, o provedor “passivo” é o que menos consome energia e sua precisão é variada. Este provedor aguarda que outra aplicação requirite a localização do aparelho para reaproveitar essa informação. Observe ainda que cada provedor possui seu próprio perfil de consumo e precisão na localização calculada do aparelho.

As regras relacionadas às transições entre os modelos de consumo de energia por parte do GPS são implementadas em *GPSManager*. As variáveis inteiras *TIME_FREQ*, *DIST_FREQ* representam os valores de tempo e distância em consumo normal e *LC_TIME_FREQ* e *LC_DIST_FREQ* representam os valores de tempo e distância em consumo reduzido. O filtro de tempo determina o intervalo de tempo mínimo, em segundos, em que o GPS irá operar e o filtro de distância indica a distância mínima, medida em metros, que deverá ser percorrida para que o sensor informe a aplicação a nova localização calculada. Assim, os métodos e atributos mais relevantes da classe *GPSManager* são apresentados na Fig. 4, sendo estes utilizados para alterar o perfil de consumo energético do GPS.

GPSManager

```

TIME_FREQ: int
DIST_FREQ: int
LC_TIME_FREQ: int
LC_DIST_FREQ: int

nextCall(): void
switchToLowerConsumption(): void
stopLowerConsumption(): void
    
```

Fig. 4. Classe *GPSManager*.

O pseudocódigo para solicitação de uma coordenada de posicionamento é apresentado a seguir:

```

nextCall:
provider ← LocationProvider.get()
se Criteria.get() = NORMAL
    c ← LocationManager.requestLocationUpdates(provider,
        TIME_FREQ, DIST_FREQ)
senão
    c ← LocationManager.requestLocationUpdates(provider,
        LC_TIME_FREQ, LC_DIST_FREQ)
    
```

Essa abordagem oferece maior flexibilidade para alterar a frequência de uso do sensor de GPS através da reinicialização do serviço utilizando os filtros de tempo e distância apropriados, segundo seu modo de consumo. A classe *Criteria* define regras para a transição de estados de consumo do GPS. Uma regra, por exemplo, pode considerar os níveis de carga da bateria para efetivar a transição para modo de consumo reduzido, quando a reserva de energia cair para níveis pré-definidos. Outra regra pode definir o uso de GPS para consumo normal quando a aplicação estiver em uso e alternar para consumo reduzido quando a aplicação estiver executando em segundo plano. A Tabela I apresenta a relação das regras consideradas durante o desenvolvimento da aplicação juntamente com seus respectivos modos de consumo de energia.

TABELA I RELAÇÃO ENTRE AS REGRAS ESTABELECIDAS COM SEUS RESPECTIVOS MODOS DE CONSUMOS

Regra	Modo de Consumo
Bateria acima de 60%	NORMAL
Bateria abaixo de 60%	REDUZIDO
Aparelho carregando	NORMAL
Aplicativo em segundo plano	REDUZIDO

O sistema operacional Android possui uma série de eventos relacionados ao estado da bateria que podem ser tratados pela aplicação. Um desses eventos é disparado no momento em que o aparelho é conectado a uma fonte de energia. Outro tipo de evento é disparado quando a bateria alcança níveis criticamente baixos ou quando a mesma estiver completamente carregada. O único evento que representa o meio termo do nível da bateria é disparado no momento em que a mesma deixa o estado crítico. Por ser um valor muito baixo (15%), optou-se por definir o nível de 60% como limiar para a transição dos estados de consumo do GPS, inspirado em Roy et al. [3]. Estas regras estabelecidas para o modo de consumo são exatamente aquelas consolidadas na Tabela 1.

A redução do acionamento dos sensores gera impactos mensuráveis para as aplicações do usuário. Estes impactos devem ser avaliados pelo construtor da aplicação que eventualmente emprega a abordagem aqui descrita, optando por tempos de atraso no acionamento compatíveis com sua aplicação. Por exemplo, considerando o sensor GPS e uma aplicação voltada para um dispositivo sendo deslocado na velocidade de passeio à pé (5 Km/h) e um atraso significativo de 10s, teria sido percorrido um espaço de 13,8m sem uma nova atualização de posição. Um dispositivo a 100 Km/h, teria sido deslocado 277,7m no mesmo período. A modelagem do

software, que por ventura esteja sendo construído, deve avaliar se estes valores são relevantes para o seu modelo de negócio.

IV. AVALIAÇÃO EXPERIMENTAL

Foram realizados testes experimentais com o objetivo de avaliar o comportamento de consumo de energia por parte dos sensores de Wifi e GPS após a incorporação das classes *NetworkScheduler* e *GPSManager* ao código da aplicação. Foram utilizados três dispositivos para a execução dos experimentos: (1) Sony Xperia Z, 1.5GHz Quad-Core, 2GB de RAM, Android 5.1.1; (2) Samsung Galaxy S3, 1.4Ghz Quad-Core, 1GB de RAM, Android 4.3; (3) Samsung Galaxy S5 mini, 1.4Ghz Quad-Core, 1.5GB de RAM, Android 5.1.1. Apesar dos dispositivos apresentarem configurações distintas, o que favorece a pluralidade no experimento, todos possuem o mesmo sistema operacional. A opção pelo sistema Android se deu por duas razões: no momento do experimento tal sistema operacional representava 73,39% da fatia de mercado mundial, 86,26% do mercado na América Latina [18]; o desenvolvimento para a plataforma segunda colocada no mercado é proprietário e pago. Assim, ao mesmo tempo que se realiza um experimento compatível com um maior número de dispositivos, há uma escolha ideológica dos autores em evitar ambientes com aprisionamento tecnológico.

Para realizar a medição do consumo de energia durante a execução dos experimentos, foi utilizado o aplicativo PowerTutor [11]. A medição do consumo por parte dos sensores de Wifi e GPS foi obtida ao expor a aplicação a longos períodos de execução contínua mantendo apenas um dos sensores ativados de cada vez. O encerramento da medição acontece quando toda a bateria do dispositivo móvel é drenada.

Cabe aqui mencionar uma restrição imposta ao desenvolvimento do trabalho, em especial à execução desta avaliação experimental. Os três dispositivos móveis empregados são celulares de mercado, habilitados na mesma operadora. Eles não foram colocados em modo off-line para evitar interferências na medição de consumo de energia proveniente de eventuais demandas da rede de telefonia celular porque isso inviabilizaria a característica de GPS assistido. Além disso, o próprio sistema operacional do celular pode disparar algum processo para ser executado no dispositivo durante a medição de consumo de energia. Isto certamente interfere na medição e não pode ser gerenciado no experimento. Desta forma, o modo encontrado para mitigar estes problemas foi realizar, para cada medição, quinze processos completos de monitoramento do consumo. O maior e o menor tempo de uso da bateria foram descartados, sendo a medição o resultado de uma média aritmética dos treze consumos de bateria restantes. Assim, buscou-se atenuar eventuais interferências externas.

Com relação ao sensor Wifi, foram realizadas 10 medições, cada qual para um tempo de atraso específico, permitindo obter o comportamento de consumo por parte do sensor nesse lapso de tempo. Essa variável é responsável por definir o tempo máximo, em segundos, que os serviços de rede devem aguardar na fila até serem executados. A Tabela II apresenta os resultados desse experimento.

TABELA II RELAÇÃO DO CONSUMO MÉDIO E DESVIO PADRÃO DO WIFI EM FUNÇÃO DO TEMPO DE ATRASO

Medição	Consumo (J)					
	Xperia Z		Galaxy S3		Galaxy S5 mini	
	Normal	Reduzido	Normal	Reduzido	Normal	Reduzido
1	403,8	60,45	375,94	57,19	365,44	55,13
2	296,48	49,77	281,66	50,58	269,50	46,53
3	245,163	26,48	231,43	25,34	222,12	28,99
4	208,56	37,44	196,88	35,23	191,67	35,04
5	170,15	32,21	163,34	29,99	154,50	29,44
6	167,64	46,04	159,93	43,46	155,91	41,76
7	145,36	29,62	136,78	27,78	131,84	30,13
8	120,43	17,87	113,93	16,80	112,00	16,35
9	110,33	21,89	104,81	22,53	99,96	19,96
10	106,06	22,17	101,29	21,22	96,41	20,17
Média	168	30	161	28	155	29
±Desvio	±94	±13	±88	±13	±85	±12
Sobrevida	5,60		5,75		5,34	

Considerando os valores obtidos listados na Tabela 2, optou-se por definir o valor de 5 segundos de intervalo para a variável de tempo de atraso. Os critérios utilizados para a escolha desse valor consideram o consumo médio, o desvio padrão e o tempo de respostas dos serviços de rede. Sob a ótica do consumo, o ideal é utilizar um atraso de 10s, porém intervalos muito longos impactam negativamente na interação do usuário com a aplicação, uma vez que seria perceptível o atraso na atualização dos dados oriundos da rede. Por outro lado, intervalos muito curtos aumentariam a interação, mas contribuiriam modestamente para a economia de energia.

Percentualmente, a diferença de consumo de energia entre cenários com atrasos de t segundos e $t+1$ segundos varia entre uma ou duas dezenas de pontos percentuais. Isto não ocorre com os atrasos de 5 e 6 segundos, quando a diferença passa a ser de dois ou menos pontos percentuais. Assim, os resultados com atrasos de 5 e 6 segundos sugerem um marcador de comportamento e estes dois valores são candidatos a pontos de corte. O atraso de 5 segundos determina um tempo menor de espera do sensor, e faz com que o sensor interaja mais frequentemente com a rede. Sendo assim, este é o valor de atraso escolhido para garantir um equilíbrio entre economia de energia e impacto na experiência do usuário.

Em seguida, um teste similar foi realizado para medir o consumo de energia por parte do sensor de GPS. Foram realizadas 10 medições, cada qual com duração de 1 hora ininterrupta. Os celulares foram deslocados em vias urbanas de tráfego rodoviário na cidade do Rio de Janeiro, sempre realizando o mesmo trajeto, em dias de semana e nos mesmos horários. A variável de tempo foi definida com 5 segundos, a variável distância foi associada a 41 metros e o nível de bateria para acionamento do modo reduzido de consumo foi definido com 60%. Os resultados desse experimento podem ser observados na Tabela III.

TABELA III RELAÇÃO DO CONSUMO MÉDIO E DESVIO PADRÃO DO GPS EM FUNÇÃO DO VALOR DO FILTRO DE TEMPO

Medição	Cenários de Consumo Médio (J)					
	Xperia Z		Galaxy S3		Galaxy S5 mini	
	Normal	Reduzido	Normal	Reduzido	Normal	Reduzido

1	4743	977	4420	918	4416	914
2	4829	1009	4539	940	4501	926
3	5124	1040	4904	960	4642	954
4	5733	931	5372	880	5320	852
5	6062	962	5686	894	5535	888
6	4354	1088	4162	1024	4041	985
7	4790	1103	4440	1022	4488	1004
8	5378	902	5114	833	4867	837
9	4619	999	4328	946	4268	919
10	5080	989	4765	945	4669	899
Média	4954	994	4773	936	4675	918
±Desvio	±525	±64	±493	±59	±459	±53
Sobrevida	4,98		5,10		5,09	

Nota-se uma redução significativa no consumo de energia da aplicação ao ser executada ao longo deste período de teste. A sobrevida da bateria foi calculada relacionando o consumo de energia normal com o consumo reduzido. O cenário de consumo reduzido indica a possibilidade de utilizar o dispositivo móvel cerca de cinco vezes mais do que o cenário de consumo normal, algo que é sustentado pelo cálculo da sobrevida nas Tabelas II e III. É importante mencionar que esse cenário de consumo reduzido é aplicado quando o aplicativo encontra-se em segundo plano ou quando a bateria possui menos de 60% de carga, conforme as regras expostas na Tabela I. Sob esta ótica, seria possível obter resultados ainda melhores se estas configurações de consumo reduzido fossem aplicadas em todas as situações.

Os resultados obtidos pelas estratégias para gerenciamento energético dos sensores de *smartphones* indicam uma redução significativa do consumo de energia. A sobrevida obtida através das estratégias de gerenciamento energético é da ordem de cinco vezes mais para ambos os sensores, sendo o ganho no caso do Wifi ligeiramente maior que do GPS. Estes valores corroboram com as percepções de Carrol e Heiser [2], Naik e Chavan [7] e Chavez et al. [12] os quais já haviam indicado certos sensores como os grandes agentes redutores do tempo de autonomia dos *smartphones*. Entretanto, não havia sido quantificada esta redução de consumo, possivelmente por conta da ausência de estratégias de gerenciamento destes sensores, tal como os propostos neste trabalho.

Uma métrica importante a ser avaliada é o coeficiente de variação (desvio padrão/média) calculável para cada coluna das Tabelas II e III. A Tabela II apresenta coeficientes de variação em torno de 50%, enquanto que a Tabela III apresenta valores inferiores a 10%. Isso indica que o comportamento do sensor GPS gerou medidas homogêneas, enquanto o comportamento do sensor Wifi produziu medidas heterogêneas, ainda que se tenha adotado uma abordagem estatística para a obtenção de cada medida. Isto remete novamente à impossibilidade de se controlar por completo o comportamento do sistema operacional e das demais aplicações instaladas no dispositivo, tal como mencionado anteriormente. Para o caso do sensor GPS, onde o coeficiente de variação foi baixo, entende-se que o ecossistema (sistema operacional + aplicativos) dos dispositivos demanda o sensor de forma mais sistemática. No caso do sensor Wifi, o coeficiente é alto, caracterizando uma grande dispersão, o que

aponta uma demanda pelo sensor mais diversa e aleatória. De fato, o acesso à rede de dados através de Wifi é bastante incerto, quando não se controla todos os possíveis geradores de tráfego. Soma-se a isto o fato de que não é apenas o ecossistema do dispositivo que aciona o sensor Wifi, mas também todo datagrama que circula na rede wireless e que chega à camada física do modelo TCP/IP implementado pela interface de rede do dispositivo.

V. CONCLUSÕES

Neste trabalho foi apresentado uma estratégia de desenvolvimento de software para restringir o uso dos sensores de dispositivos móveis a fim de reduzir a sua demanda por energia. Através dos experimentos realizados foi possível comprovar que a estratégia proposta permitiu a redução do consumo de energia controlando a frequência de uso dos sensores. Também foi possível determinar a escolha dos valores mais apropriados para o tempo de atraso entre as requisições do sensor Wifi. Em seguida, foi realizado com sucesso um experimento comparativo dos perfis de consumo de energia de um dispositivo que não fazia o uso da estratégia de consumo para o sensor GPS e outro que o fazia.

Através dos dados obtidos foi possível notar um ganho significativo na redução do consumo de energia por parte da aplicação após a adoção das estratégias do atraso para uso do sensor de Wifi e da definição de intervalos para uso do GPS.

A taxa de atualização dos sensores foi definida considerando uma relação de consumo de energia pelo sensor em relação ao impacto na experiência do usuário. Uma proposta para trabalho futuro poderia visar o aumento da confiança dos dados adquiridos acerca da percepção do usuário e quantificá-los utilizando metodologias formais, como por exemplo, utilizando a lógica nebulosa. No caso do GPS, isto resultaria em maior precisão na classificação dos valores da taxa de atualização de acordo com as três categorias de percepção do usuário.

No contexto do desenvolvedor de aplicações para dispositivos móveis há uma grande variedade de certificações que validam seus conhecimentos acerca de diversos parâmetros, tais como o sistema operacional e suas particularidades, a construção de interfaces intuitivas e a segurança do armazenamento e manipulação dos dados do usuário. No entanto, as certificações não avaliam o consumo de bateria por parte das aplicações como um fator relevante e, portanto, o mesmo tende a ser deixado de lado. Na concepção dos autores deste artigo, esta prática de mercado está equivocada. Os resultados obtidos por esse trabalho demonstram que a preocupação em relação ao consumo de energia de aplicativos durante o processo de desenvolvimento irá auxiliar na preservação do tempo de vida útil da bateria em longo prazo e consequentemente uma maior disponibilidade do equipamento.

VI. AGRADECIMENTOS

Esta pesquisa foi apoiada pelo Fundo Poli 19.257 da Fundação Coppetec da Universidade Federal do Rio de Janeiro (UFRJ). O trabalho foi desenvolvido no Laboratório de Inteligência de Máquina e Modelos de Computação (IM²C) da Escola Politécnica (Poli) da UFRJ.

REFERÊNCIAS

- [1] Tarkoma, Sasu; Siekkinen, Matti; Lagerspetz, Eemil; Xiao, Yu. "Smartphone Energy Consumption: Modeling and Optimization", Cambridge University Press, New York, NY, USA, 2014, pp. 76.
- [2] Carrol, A., Heiser, G., "An analysis of Power Consumption on Smartphones", in Proceedings of the 2010 USENIX, USENIXATC'10, Berkeley, 2010, p. 21.
- [3] Roy, A., Rumble, S., Stutsman, R., Levis, P., Mazieres, D., "Energy Management in Mobile Devices with the Cinder Operating System", in Proceedings of the Sixth Conference on Computer Systems, EuroSys '11, New York, 2011, pp. 139-152. doi: 10.1145/1966445.1966459
- [4] Li, Ding., Halfond, Wiliam, "An investigation into Energy-Saving programming practices for Android Smartphone App Development", in Proceedings of the 3rd International Workshop on Green and Sustainable Software, NGREENS 2014, ACM, New York, 2014, pp. 46-53. doi: 10.1145/2593743.2593750
- [5] Nedeveschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D., "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation", in Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI'08, USENIX Association, Berkeley, 2008, pp. 323-336.
- [6] Xia, F., Zhang, Wei., Ding, F., Hao, F., "A-GPS Assisted Wifi Access Point Discovery on Mobile Devices for Energy Saving", in Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization, 2009, pp. 67-76. doi: 10.1109/GIIS.2011.6026710
- [7] Ellison, A., Ellison, R., Rance, D., Greaves, S., Standen, C. Harnessing "Smartphone sensors for tracking location to support travel data collection", in 10th International Conference on Transport Survey Methods, Leura, Australia, 2014.
- [8] Naik, C., Chavan, R., "Optimization in Power Usage of Smartphones", International Journal of Computer Applications, vol. 119, no. 18, pp. 7-13, 2015. doi: 10.5120/21165-4224
- [9] Podlesny, M., "Networking Mechanisms for Delay-Sensitive Applications", PhD dissertation, Computer Science and Engineering, Washington University, St Louis, USA, 2009. doi: 10.7936/K7668B7G
- [10] Couto, M., Carção, T., Cunha, J., Fernandes, P., Saraiva, J., "Detecting Anomalous Energy Consumption in Android Applications", in Proceedings of the eighteenth Brazilian Symposium, SBLP, Maceió, 2014, pp. 77-91. doi: 10.1007/978-3-319-11863-5_6
- [11] Dick, Robert, "PowerTutor", 2018. [Online]. Available: <http://ziyang.eecs.umich.edu/projects/powertutor/>, Accessed on February, 20, 2019.
- [12] Chavez, Diego; Espinosa, Santiago; Cazco, Diego Arias. "Reactive Power Optimization of the Electric System based on Minimization of Losses". IEEE Latin America Transactions, vol. 14, no. 11, pp. 4540-4546, Nov, 2016. doi: 10.1109/TLA.2016.7795826
- [13] A. Bravo Vicente, J. Galan Jimenez and A. Gazo Cervero, "Characterization of Energy Consumption of a 802.11g Device", IEEE Latin America Transactions, vol. 13, no. 8, pp. 2495-2499, Aug. 2015. doi: 10.1109/TLA.2015.7331903
- [14] E. L. Schneider, R. de Freitas Valle Dresch, L. T. dos Reis Loureiro and R. Machado de Brito, "Analysis Method of Mobile Phone Batteries Remaining State of Charge," IEEE Latin America Transactions, vol. 15, no. 5, pp. 854-860, May 2017. doi: 10.1109/TLA.2017.7910198
- [15] N. Avelino de Santana, F. Antonio Aires Lins and E. Teixeira Gomes de Sousa, "Performance Evaluation of Mobile Applications in Mobile Cloud Environments," IEEE Latin America Transactions, vol. 14, no. 11, pp. 4597-4602, Nov. 2016. doi: 10.1109/TLA.2016.7795835
- [16] E. M. Diaz, E. T. Marquez, L. M. d. A. Recio and I. B. F. d. Murias, "SON Use Case Study "Energy Saving" for LTE eNBs," IEEE Latin America Transactions, vol. 8, no. 2, pp. 184-189, April 2010. doi: 10.1109/TLA.2010.5514446
- [17] Oliveira, Leonardo da Silva; Oliveira, Priscila da Silva, "Redução do Consumo de Energia para Apps Baseados em Geolocalização", in XIII Simpósio de Excelência em Gestão e Tecnologia, Rio de Janeiro, 2016.

StatCounter Global Stats, "Mobile Operating System Market Share Worldwide", 2019. [Online]. Available: <http://gs.statcounter.com/os-market-share/mobile/worldwide>, Accessed on February, 24, 2019.



Leonardo da Silva Oliveira realizou seu MBA em Engenharia de Computação e Sistemas pela Escola Politécnica da Universidade Federal do Rio de Janeiro – UFRJ (2016), bacharel em Ciência da Computação pela Universidade Veiga de Almeida – UVA (2014).

Profissional com mais de cinco anos de experiência em desenvolvimento de software, responsável pela manutenção de sistemas comerciais de grande impacto e visibilidade. Atualmente atua no mercado financeiro internacional utilizando e comercializando seus próprios robôs automatizados baseados em sistemas de trade consistentes.



Flávio Luis de Mello recebeu o título de DSc em teoria da Computação e Processamento de Imagens pela Universidade Federal do Rio de Janeiro - UFRJ (2006), MSc. Em Computação Gráfica pela Universidade Federal do Rio de Janeiro (2003), graduou-se em Engenharia de Sistemas pelo Instituto Militar de Engenharia - IME (1998). Desenvolveu sistemas de comando e controle, e implementou aplicações para troca de mensagens militares durante doze anos como oficial do Exército Brasileiro. Foi responsável por desenvolver aplicações baseadas em aprendizado de máquina e raciocínio dedutivo para o Grupo Mentor.

Dr Mello atualmente é professor associado no Departamento de Eletrônica e Computação (DEL) da Escola Politécnica (Poli) na Universidade Federal do Rio de Janeiro (UFRJ). Ele ocupa a chefia do Laboratório de Inteligência de Máquina e Modelos de Computação (IM2C).